Abstract

Optimal mechanisms are determined for the hierarchical decomposition of wire-frame surfaces generated by box-splines. A family of box-splines with compact support, suitable for the approximation of wire-frames is first defined, generated by arbitrary sampling matrices with integer eigenvalues. For each such box-spline, the optimal positioning of the wire-frame nodes is determined for each level of the hierarchical wire-frame decomposition. Criterion of optimality is the minimization of the variance of the error difference between the original surface and its representation at each resolution level. This is needed so as to ensure that the wire mesh produces at each resolution as close a replica of the original surface as possible. Several such combinations of box-spline generated meshes and the corresponding optimal node lattice sequences are examined in detail with a view to practical application. Their specific application to the hierarchical coding of 3D wire meshes is experimentally evaluated.

I INTRODUCTION

Hierarchical structure decomposition consists of linear transformations of the initial structure arranged in successively reduced resolution levels. In this process, large scale features and relations successively reduce
to more local operations. Further, the sequence of residues consists of largely decorrelated values which hence admit parsimonious representations. These useful properties have made hierarchical or multiresolution decomposition a favored tool in many applications including computer vision and image coding.

In this paper optimal mechanisms are investigated for multi-grid hierarchical decomposition of wireframe surfaces. Wire-frames, i.e piecewise polynomial approximations of surfaces are widely used for modeling the structure of 3D objects, in the computer vision, computer graphics and image coding areas. A multi-grid decomposition of such surfaces is in particular highly useful in the following applications:

- **Computer Graphics**: The time required to render a scene is usually proportional to the number of polygons that constitute 3D model objects. Lower detail representations of the objects may be used when the objects are located far from the viewer or when they are of limited interest. Also in browsing a large database of 3D models, a low detail preview is usually required. Finally, in computer aided design, editing at different levels of detail is useful for designing complex models [1]. All of the above may be achieved by keeping a pyramid of lower detail representations along with the original 3D model. These representations should ideally be as close replicas of the original as their lower resolution permits.

- **Computer Vision**: Coarse-to-fine transition schemes characterize many efficient computer vision algorithms that attempt to estimate 3D motion and structure of objects. Such schemes often succeed to impose global constraints to the solution and thus avoid local minima [2, 3]. Multi-grid decompositions may be used to implement such schemes, because of their capability of suppressing local detail while maintaining global surface features.

- **Compression**: The 3D surface decomposition produces a pyramid of successively lower resolution approximations of the original surface along with the errors produced in passing from one resolution to the next. By appropriately quantizing the error signals, significant compression of the surface representation may be achieved within a specified error tolerance. This is particularly useful when the surface representation is characterized by a high degree of redundancy, as is the case, for example with surfaces created by sampling dense depth maps [4].

In all of the above applications, a requirement often encountered is that the hierarchical decomposition result in lower resolution replicas retaining as much resemblance to the original as possible. This is obviously desirable for example, when building lower-resolution representations for the population of a data bank.
used for browsing. It is also desirable for efficient hierarchical compression. Indeed, in a typical scalable coding application, this high quality lower resolution replica may be transmitted via a slower communication channel while the original is perfectly reconstructed from the entire pyramid (figure 7). Likewise, in a progressive coding scheme, this will ensure optimum quality of the series of replicas produced, originally with low resolution, which gradually increases until the desired quality emerges. This makes the resulting pyramid appropriate for the hierarchical coding of wire mesh objects for applications where progressive or scalable representation of 3D objects is desirable.

Methodologies for the realization of optimal, in the above sense, hierarchical decompositions of scalar multivariable signals were recently proposed in [5] and [6]. Two distinct approaches were followed. The first aims to minimize the variance of the error image between any two successive stages of the decomposition and is described in [5]. The goal of the second and more ambitious approach proposed in [6] is to minimize the variance of the error between the original image and that produced at any one of the lower resolutions. With either method, given arbitrary analysis or synthesis filters, the optimal corresponding synthesis, respectively analysis filters are found so as to minimize the respective error variances. Both above approaches were recently seen to produce satisfactory results when used for the implementation of wavelet-based image compression schemes [9].

An elegant framework for the efficient multiresolution analysis of 3D surfaces is afforded by decomposition using two-dimensional wavelet bases. Such bases are generally constructed as tensor products of univariate wavelet bases [10]. A wavelet-based representation for multi-resolution B-spline curves was developed in [13]. The curve shape may be edited at arbitrary scales by modifying the underlying wavelet coefficients, however extension to surfaces is limited to tensor product splines. Non tensor product expansions have also been proposed [10, 12].

Mesh simplification procedures were described in [14, 15, 16], where selective removal of mesh vertices followed by retriangulation is proposed. Similar approaches are presented in [17, 18]. The above schemes try to approximate a complex mesh with a simpler one and do not allow for multi-resolution representation of the mesh. In [19] an optimization-based mesh simplification procedure is presented. This is used in [20] to construct a hierarchical representation of the mesh which is oriented to progressive transmission rather than compression and has very high computational complexity.

Wavelets based on subdivision surfaces were defined in [47] for the representation of functions defined on arbitrary two-dimensional topological domains. This scheme requires that the mesh has subdivision connectivity i.e. higher scale meshes may be constructed by iterative subdivision of a base mesh. In [21]
an algorithm that approximates an arbitrary mesh with one having subdivision connectivity is described. This is extended in [22] to include color and support progressive transmission. In [40] the “lifting scheme” is introduced, which allows building of second generation wavelets i.e. wavelets that are not necessarily translates and dilates of one function. This scheme is used in [41] to construct wavelets of scalar functions defined on the sphere. For a more complete survey of work related to multiresolution analysis of piecewise linear triangular meshes, see [42].

An important class of functions that may be used to construct wavelet bases for 3D surface meshes without the disadvantages of tensor product surfaces are box-splines. These functions and the corresponding wavelet bases were recently discussed in [11, 26]. Tensor product splines have been shown to be inappropriate for the modeling of complex objects in numerous applications due to their definition over rectangular parameter domains. Triangular splines, on the other hand, can be used to construct smooth surfaces defined over arbitrary triangulation of the parameter domain. This scheme uses basis functions and control points to form piecewise polynomial surfaces of degree $n$ that exhibit $n - 1$ continuity. This representation offers affine invariance and allows local control. Moreover, points on such a surface lie in the convex hull of the associated control points. Finally, every piecewise polynomial surface can be expressed in this form.

In the present paper, we adapt and generalize the approach in [11, 26] to parameterize a general class of wavelet bases generated by box splines, for the representation of wire-meshes. We restrict our attention to wavelet bases associated with spatially invariable filter banks. We show that each box spline may be associated with one or more sampling matrices and each defines a FIR spatially invariant synthesis filter and a synthesis scaling function for a hierarchical wire-mesh decomposition. Further, based on the results in [6] we then choose corresponding analysis filters determining the node locations so as to minimize the variance of the errors in the representation of the original by one or a series of lower resolution meshes. We use these results to construct the optimal decomposition filters for a general class of wire-frame surfaces including the popular regular triangulated domain. Experimental results are then obtained exemplifying the hierarchical decomposition of 3D surfaces using these filters.

The wavelet bases developed in this paper have several advantages over those presented in [47]. Specifically, in [47] the decomposition is produced by filters with coefficients varying over the mesh and is computationally costly since computing the analysis filters requires solution of a linear system for each vertex. Furthermore the scaling functions are always piece-wise linear, and sub-optimal in the least squares sense, while the ones presented in this paper may use a higher order box-spline basis function (which may represent complex surfaces with fewer mesh vertices) and the the resulting analysis-synthesis filters are optimal.
Since our scheme is based on the use of box splines it applies only to surfaces that are homomorphic to a plane, cylinder or torus. Its applicability is, therefore, accordingly limited. However there exist numerous applications in areas such as visualization of medical data, computer aided design and image and video coding where surfaces are represented using simple topology triangle meshes. In image and video coding in particular, which is our main concern, the meshes used are homomorphic to a plane. We finally note that even complex topological surfaces may be decomposed into surface patches with simple topology [45] (although, preserving the continuity and smoothness of a mesh consisting of several patches, as the scale changes, is a highly nontrivial undertaking).

The use of box-splines for the optimal in the above sense hierarchical representation of wire-mesh surfaces is described in section II. Examples of the resulting optimal wavelet bases are derived in section III. Experimental results are presented in section IV. Conclusions are finally drawn in section V.

II USE OF BOX-SPLINES FOR THE OPTIMAL HIERARCHICAL REPRESENTATION OF WIRE-MESH SURFACES

A wire-frame is a surface consisting of adjacent patches of various shapes. This surface may be defined as a parametric function \( P(u, v) = [x(u, v), y(u, v), z(u, v)]^T \) in the parametric space \( U \subseteq \mathbb{R}^2 \), determined by the position of a set of control points or nodes \( r[k, l] = [x_{kl}, y_{kl}, z_{kl}]^T \). This representation is more flexible than an explicit one of the form \( z = f(x, y) \) since it allows for arbitrary, possibly closed wire-frame surfaces to be defined. For example a wire-frame approximation of a sphere

\[
\begin{bmatrix}
  x(u, v) \\
  y(u, v) \\
  z(u, v)
\end{bmatrix}
= \begin{bmatrix}
  \rho \cos u \cos v \\
  \rho \cos u \sin v \\
  \rho \sin v
\end{bmatrix}
\]

is defined by a wire frame (e.g. a triangulation) of the parametric space \( U = [(u, v) : -\pi \leq u, v < \pi] \).

An important class of functions for the hierarchical representation of wire-meshes are multivariate box-splines. These were studied in [25]. More recent surveys may be found in [26, 27, 28]. We shall use the following definition [11, 26] for a box spline

**Definition 1** A bivariate box-spline \( \tilde{\psi}(u, v) \) on \( \mathbb{R}^2 \) is defined by \( \mu \) not necessarily distinct vectors \( (e_1, e_2, \ldots, e_\mu) \)
with $\varepsilon_1$ and $\varepsilon_2$ linearly independent. Its support $X_\mu$ is compact:

$$X_\mu = \{ \rho_1 \varepsilon_1 + \ldots + \rho_\mu \varepsilon_\mu, \ 0 \leq \rho_i \leq 1, \ 1 \leq i \leq \mu \}, \quad (1)$$

and the box spline is a piecewise polynomial defined by

$$\tilde{\psi}(u, v) = Q(u, v|Y_\mu) \quad (2)$$

where $Y_\mu$ is the matrix

$$Y_\mu = [\varepsilon_1 \ldots \varepsilon_\mu]$$

and $Q$ is a function defined recursively by

$$Q(u, v|Y_\mu) = \int_0^1 Q(u - \tau e_{\mu x}, v - \tau e_{\mu y}|Y_{\mu-1}) d\tau$$

where $e_\mu = [e_{\mu x} \ e_{\mu y}]^T$, and

$$Q(u, v|Y_2) = \begin{cases} 1/|\det Y_2| & \text{if } (u, v) \in X_2 \\ 0 & \text{otherwise} \end{cases}$$

The Fourier Transform of the box spline (2) equals [11, 26]:

$$\tilde{\Psi}(\omega) = \prod_{k=1}^{\mu} \frac{e^{j\omega^T e_{k-1}}}{j\omega^T e_k} \quad (3)$$

where $\omega = [\omega_1, \omega_2]^T$. It may be seen [26] that $\tilde{\psi}(u, v)$ satisfies a two-scale relation and generates a corresponding multiresolution analysis based on a mesh subdivision by the sampling matrix $M = \text{diag}\{2, 2\}$. For purposes of hierarchical coding, however, it is important that $M$ be allowed to be nondiagonal.

In fact the use of nondiagonal sampling matrices such as quincunx and the hexagonal matrix is known to often lead to superior results in signal and image coding. Further, the matrix $M = \text{diag}\{2, 2\}$ only allows a 4:1 change of scale in each step. As will be seen in the experimental results, better compression may be achieved with a 2:1 scale change which is only possible with nondiagonal $M$. For these reasons, we shall develop in this section a novel biorthogonal wavelet basis generated by box splines with nondiagonal...
sampling function $M$. To do this we shall let $M$ be any matrix such that for each $k$, $k = 1, \ldots, \mu$,

$$M e_k = \lambda_k e_k, \quad \lambda_k : \text{integer} \quad (4)$$

or, for pairs $(k, l)$

$$M e_k = \lambda_k e_l, \quad M e_l = \lambda_l e_k, \quad \text{hence} \quad M^2 e_k = \lambda_k \lambda_l e_k$$

or else, for triplets $(k, l, m)$

$$M e_k = \lambda_k e_l, \quad M e_l = \lambda_l e_m, \quad M e_m = \lambda_m e_k, \quad \text{hence} \quad M^3 e_k = \lambda_k \lambda_l \lambda_m e_k$$

and so on. Thus the vectors $e_k$ are eigenvectors of integer powers $M^n$ of $M$ with integer eigenvalues $\lambda_k$.

Then,

$$\tilde{\Psi}(M^T \omega) = \prod_{k=1}^{\mu} \frac{1}{\lambda_k} \frac{e^{j\lambda_k \omega^T e_k} - 1}{e^{j\omega^T e_k} - 1} \tilde{\Psi}(\omega) \quad (5)$$

For integer $\lambda_k$ the above right hand-side of (5) will obviously be a finite-order polynomial in $e^{j\omega^T e_k}$. It follows that if $e_k$ have integer entries, the function $\tilde{\psi}(\mathbf{t}) = \tilde{\psi}(u, v)$, where $\mathbf{t} = [u, v]^T$, will satisfy a two-scale difference equation of finite order

$$\tilde{\psi}(\mathbf{t}) = \sqrt{M} \sum_{k} g(k) \tilde{\psi}(M \mathbf{t} - k) \quad (6)$$

where, $M = \text{det} M$. From (5), (6) the Fourier transform of $g(k)$ will in fact be

$$G(\omega) = \sum_k g(k) e^{j k^T \omega} = \sqrt{M} \prod_{k=1}^{\mu} \frac{1}{\lambda_k} \frac{e^{j\lambda_k \omega^T e_k} - 1}{e^{j\omega^T e_k} - 1} \quad (7)$$

Clearly, $|G(\omega)|^2 = G(\omega)G(-\omega)$ will be a finite-order polynomial in $e^{j\omega^T e_k}$. Note that from (5,7)

$$\tilde{\Psi}(\omega) = \frac{1}{\sqrt{M}} G(M^{-T} \omega) \tilde{\Psi}(M^{-T} \omega) = \prod_{s \geq 1} \frac{1}{\sqrt{M}} G((M^T)^{-s} \omega)$$

where $M^{-T}$ is a shorthand notation for $(M^T)^{-1}$.

The cardinal box-spline surface family generated by the box-spline $\tilde{\psi}(\mathbf{t})$ was studied in [11, 26] for the
special case of $M = \text{diag\{2, 2\}}$. In general, this surface is described by

$$P_s(t) = \sum_k r_s[k] \tilde{\psi}(M^s t - k)$$

(8)

where the scale is indicated by the nonnegative integer $s$, $P = [x, y, z]^T$ is a point in $R^3$ and $r_s$ are the nodes of the wire-frame. A biorthogonal wavelet basis for the representation of a 3D function $P(t)$ is defined by the dual relationship

$$r_s[k] = \int P(t) \psi(M^s t - k) dt$$

(9)

where

$$\psi(t) = \sqrt{M} \sum_k h[k] \psi(M t - k)$$

(10)

and

$$P(t) = \lim_{s \to \infty} P_s(t).$$

(11)

For example, in [29] for the special case $M = \text{diag\{2, 2\}}$, conditions are given under which the support of $h[k]$ is bounded and (11) is valid in the $L^2$ sense. Other conditions under which the above construction is valid have been given in [27, 28, 30] again, only for the case where $M = \text{diag\{2, 2\}}$, and in [6] for arbitrary $M$. It will be useful for the subsequent development to note that in any one of these settings, (9) and (10) immediately imply

$$r_s[n] = \sqrt{M} \sum_k h[k] r_{s+1}[M n - k]$$

(12)

meaning that the node lattice at resolution $s$ is produced by filtering the node lattice at resolution $s + 1$ using a filter with impulse response $h[k]$ and subsampling the result with sampling matrix $M$. This is shown schematically in figure 1. Also,

$$\Psi(\omega) = \frac{1}{\sqrt{M}} H(-M^{-T} \omega) \Psi(M^{-T} \omega) = \prod_{s \geq 1} \frac{1}{\sqrt{M}} H((M^T)^{-s} \omega).$$
As mentioned in the Introduction, in many applications of hierarchical wire mesh decomposition, it is desirable to minimize a measure of the approximation error \( P(u, v) - P_s(u, v) \) at resolution \( s \) of the decomposition. In this way, the output of the decomposition will, at a specific resolution, approximate optimally the input. This is needed in applications such as image classification and hierarchical (progressive or scalable) image coding, where a version of the original mesh is needed, which at a lower resolution retains as much fidelity to the original as possible. This minimization of the approximation error was investigated in [6]. Specifically, it was shown that if \( P(u, v) = [x(u, v), y(u, v), z(u, v)] \), \( P_s(u, v) = [x_s(u, v), y_s(u, v), z_s(u, v)] \) and if \( x(u, v) \) is wide-sense stationary [43], the limit

\[
E_s^x = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} \int_{-T}^{T} E\{|x(u, v) - x_s(u, v)|^2\} dudv
\]

always exists and defines a useful measure of the convergence of \( x_s(u, v) \) to \( x(u, v) \). Likewise errors \( E_s^y, E_s^z \) may be defined, with a total error

\[
E_s = E_s^x + E_s^y + E_s^z = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} \int_{-T}^{T} E\{|P_s(u, v) - P(u, v)|^2\} dudv .
\] (13)

The minimization of this error is studied in [6], where it is assumed that arbitrary \( G(\omega) \) are given, satisfying the conditions

\[
G(0) = \sqrt{M}
\] (14)

\[
G(2\pi M^{-T} q_i) = 0, \ i = 1, \ldots, M - 1
\] (15)

where \( M = \text{det} M \), \( M^{-T} \) is a shorthand notation for \( (M^T)^{-1} \) and \( q_i \), \( i = 1, \ldots, M - 1 \) are the nonzero coset vectors of \( M^T \) [8]. It is then shown that each of the above \( E_s^x, E_s^y, E_s^z \), hence also the above overall error \( E_s \) at any one resolution \( s \), is minimized with the choice of analysis sampling function

\[
\Psi(\omega) = \frac{\tilde{\Psi}(\omega)}{A(\omega)}
\] (16)

where

\[
A(\omega) = \sum \tilde{\Psi}(\omega + 2\pi m)\tilde{\Psi}(-\omega - 2\pi m).
\] (17)

Equivalently, as shown in [6], the error (13) is minimized by choosing the generating perfect reconstruction
filter bank with its synthesis filter and analysis filter related by

\[ H(-\omega) = \frac{G(\omega)A(\omega)}{A(M\omega)} \quad (18) \]

the resulting minimum error corresponding to the x-component is given by

\[ (2\pi)^2 E_x^s = M^s \int_\Omega \Phi_x \left( (M^T)\omega \right) \left[ 1 - \frac{\tilde{\Psi}(-\omega)\Psi(\omega)}{A(\omega)} \right] d\omega \quad (19) \]

and the total error is

\[ (2\pi)^2 E_s = M^s \int_\Omega [\Phi_x \left( (M^T)\omega \right) + \Phi_y \left( (M^T)\omega \right) + \Phi_z \left( (M^T)\omega \right)] \left[ 1 - \frac{\tilde{\Psi}(-\omega)\Psi(\omega)}{A(\omega)} \right] d\omega \quad (20) \]

where

\[ \Omega = \{ \omega : |\omega_i| < \pi, \quad i = 1, 2 \}. \quad (21) \]

The error, therefore, depends explicitly on the matrix \( M \) used and not solely on the functions \( \Psi(\omega) \) and \( A(\omega) \) it generates. In this way the optimal mesh subdivision is found, so that for arbitrary resolution \( s \), the mesh \( P_s(u,v) \) approximates optimally the surface \( P(u,v) \). The specific implementation of the mesh subdivision depends on the choice of the underlying box splines.

We summarize the results of this section in the form of a concluding theorem

**Theorem 1** Let \( P(t) = [x(t), y(t), z(t)]^T, \ t \in \mathbb{R}^2 \) be a three-dimensional wide-sense stationary process, \( \tilde{\psi}(t) \) a box-spline and \( M \) a matrix satisfying (4). Then:

i There exists a finite-order polynomial synthesis filter transfer function \( G(\omega) \) satisfying (7) and generating the M-scale relation (6) for \( \tilde{\psi}(t) \).

ii If \( G(\omega) \) also satisfies (14,15), and if the analysis filter transfer function is chosen by (18), the sequence (8) of wire-mesh surfaces \( P_s(t) \) approaches \( P(t) \) as \( s \to +\infty \) in the sense of producing a vanishing error sequence (13).

iii Given any \( G(\omega) \) specified by (7) and satisfying (14,15), the choice (18) for the analysis filter minimizes the corresponding approximation error (13) for each and every scale \( s \). The minimum value of this error is given by (20).

In the remainder of this paper, examples of several such meshes typical of those most frequently used in
practice are analyzed, and their compression performance for the progressive transmission of wire-mesh surface approximations is evaluated.

III SPECIFIC WIRE MESH HIERARCHICAL DECOMPOSITIONS.

The algorithms developed and evaluated in this section are the prime candidates for use for coding purposes since they are based on sampling matrices that are known to produce good compression results for two-dimensional signals. We shall also experiment with different ways of selecting the basis vectors $e_i$ so as to evaluate the efficiency of the resulting algorithms when used for the hierarchical coding of 3D meshes.

Algorithm 1

If the matrix $M$ corresponding to quincunx sampling [7], [8] is used to generate the synthesis filters:

$$M = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}$$

then from (4)

$$\lambda_1 = 2, \quad e_1 = [1 \ 0]^T; \quad \lambda_2 = 1, \quad e_2 = [-1 \ 1]^T$$

From (7)

$$G(\omega) = \sqrt{2} \prod_{k=1}^{2} \frac{1}{\lambda_k} \frac{e^{j\lambda_k \omega T} e_k - 1}{e^{j\omega T} e_k - 1} = \frac{1}{\sqrt{2}} (e^{j\omega_1} + 1).$$

This is immediately seen to satisfy the necessary conditions (14,15) since in this case $q_1 = [1 \ 1]^T$ and $2\pi M^{-T} q_1 = [\pi \ \pi]^T$. It is then easily seen [39] that in this case $A(\omega) = 1$ and the optimal biorthogonal wavelet basis reduces to the orthonormal Haar basis:

$$H(\omega) = G(-\omega) = (e^{-j\omega_1} + 1)/\sqrt{2}.$$  

The same result is easily seen to be obtained if the different quincunx realization matrix is used

$$M = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}$$

since in this case

$$M e_1 = e_2, \quad M e_2 = 2e_1, \quad i.e. \quad M^2 e_i = 2e_i, \quad i = 1, 2$$
where
\[ e_1 = [0 \ 1]^T, \ e_2 = [1 \ 1]^T. \]

Algorithm 2

If the hexagonal sampling matrix is used to generate the synthesis filter:

\[ M = \begin{bmatrix} 2 & 1 \\ 0 & -2 \end{bmatrix} \]

we find \( \lambda_1 = 2, \ e_1 = [1 \ 0]^T, \ \lambda_2 = -2, \ e_2 = [0 \ 1]^T. \) In this case, from (7)

\[
G(\omega) = \frac{\sqrt{4} e^{j2\omega_1} e_1 - 1}{\sqrt{4} e^{j2\omega_1} e_2 - 1} = \frac{1}{2} (e^{j\omega_1} + 1)(e^{j\omega_2} + 1)e^{-2j\omega_2}.
\]

The necessary conditions (14, 15) are satisfied because

\[
2\pi M^{-T}q_1 = [0 -\pi]^T, \ 2\pi M^{-T}q_2 = [\pi \pi/2]^T, \ 2\pi M^{-T}q_3 = [\pi -\pi/2]^T.
\]

Thus, from (3)

\[
\tilde{\Psi}(\omega_1, \omega_2) = e^{-2j\omega_2} e^{j\omega_1/2} \frac{\sin^{\omega_1/2}}{\omega_1/2} e^{j\omega_2/2} \frac{\sin^{\omega_2/2}}{\omega_2/2}.
\]

To calculate the infinite sum

\[
A(\omega_1, \omega_2) = \sum_{m_1} \sum_{m_2} \tilde{\Psi}(\omega_1 + 2\pi m_1, \omega_2 + 2\pi m_2) \tilde{\Psi}(\omega_1 - 2\pi m_1, \omega_2 - 2\pi m_2) = \sin^2(\frac{\omega_1}{2}) \sin^2(\frac{\omega_2}{2}) B(\frac{\omega_1}{2}, \frac{\omega_2}{2})
\]

where

\[
B(x_1, x_2) = \sum_{m_1} \sum_{m_2} \frac{1}{(x_1 + \pi m_1)^2 (x_2 + \pi m_2)^2}
\]

we note that [39]

\[
\sum_r \frac{1}{(\omega + \pi r)^{2N}} = \frac{(-1)^{N}}{(2N - 1)!} \frac{d^{2N-1}}{d\omega^{2N-1}} \cot \omega.
\]

This gives

\[
B(x_1, x_2) = \frac{d \cot x_1}{dx_1} \left( - \frac{d \cot x_2}{dx_2} \right) = \frac{1}{\sin^2 x_1 \sin^2 x_2}
\]

and hence \( A(\omega_1, \omega_2) = 1. \) In this case the biorthogonal wavelet base reduces to an orthonormal one and the
optimal analysis filter is

\[ H(\omega_1, \omega_2) = G(-\omega_1, -\omega_2) = \frac{1}{2}(1 + e^{-j\omega_1})(1 + e^{-j\omega_2})e^{2j\omega_2}. \]

Greater freedom in selecting the box spline family is afforded by matrices which are arbitrary integer multiples of the identity matrix. For example if

\[
M = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}
\]

equation (4) is satisfied for arbitrary \( \varepsilon_k \) if \( \lambda_k = 2 \). Note however that the necessary conditions (14, 15) need to be satisfied. Since

\[
q_1 = [1 \ 0]^T, \quad q_2 = [0 \ 1]^T, \quad q_3 = [1 \ 1]^T,
\]

these require

\[
G(0, \pi) = G(\pi, 0) = G(\pi, \pi) = 0
\]

for which it suffices to have

\[
\varepsilon_1 = [1 \ 0]^T, \quad \varepsilon_2 = [0 \ 1]^T.
\]

(27)

The three algorithms that follow will utilize the above “separable” sampling matrix. In this case, the general expressions (16-18) simplify to:

\[
\Psi(\omega_1, \omega_2) = \frac{\tilde{\Psi}(\omega_1, \omega_2)}{A(\omega_1, \omega_2)}
\]

where

\[
A(\omega_1, \omega_2) = \sum_{m_1} \sum_{m_2} \tilde{\Psi}(\omega_1 + 2\pi m_1, \omega_2 + 2\pi m_2)\tilde{\Psi}(-\omega_1 - 2\pi m_1, -\omega_2 - 2\pi m_2)
\]

and

\[
H(-\omega_1, -\omega_2) = \frac{G(\omega_1, \omega_2)A(\omega_1, \omega_2)}{A(2\omega_1, 2\omega_2)}.
\]

Algorithm 3

Assume \( \mu = 2 \), with the above \( M = \text{diag}\{2, 2\} \). Then

\[
G(\omega_1, \omega_2) = \frac{1}{2}(1 + e^{j\omega_1})(1 + e^{j\omega_2}) = 2e^{j\omega_1/2} \cos(\frac{\omega_1}{2})e^{j\omega_2/2} \cos(\frac{\omega_2}{2})
\]
which is essentially identical to the synthesis filter found in example 2. Proceeding precisely as with Algorithm 2 we find $A(\omega_1, \omega_2) = 1$. Thus in this case as well, the biorthogonal wavelet base reduces to an orthonormal one and the optimal analysis filter is

$$H(\omega_1, \omega_2) = G(-\omega_1, -\omega_2) = \frac{1}{2} (1 + e^{-j\omega_1})(1 + e^{-j\omega_2}).$$

In the above case the parametric space $U$ is tessellated with rectangular tiles with side equal to one. The corresponding scaling function $\tilde{\psi}(u, v)$ is the rectangular pulse:

$$\tilde{\psi}(u, v) = \begin{cases} 1 & 0 \leq u, v < 1 \\ 0 & \text{otherwise} \end{cases}.$$

Note that in this case, the analysis filter is also of the form (7), with $e_1 = [0 \ -1]^T$, $e_2 = [-1 \ 0]^T$, i.e. the analysis sampling function is also generated by a box spline.

Finally, it must be noted that the errors produced by Algorithms 2 and 3 are not equal. In fact from (19),(20) it is seen that even though the respective products $\tilde{\Psi}(\omega)\tilde{\Psi}(-\omega)$ and the functions $A(\omega)$ are identical in these two algorithms, the terms $\Phi_x((M^T)^n\omega)$ differ. If, as is sometimes assumed for images [35], the corresponding correlation function is of the form $R_x[m_1, m_2] = \alpha \mu^{\max{|m_1|,|m_2|}}$ where $\alpha > 0, \mu < 1$, it is easily seen from (19) that the error produced by Algorithm 2 is always smaller than that of Algorithm 3.

Algorithm 4

With $M = \text{diag}(2, 2)$, if $\mu = 4$ and $e_3 = e_1$, $e_4 = e_2$ with $e_1, e_2$ given by (28),

$$G(\omega_1, \omega_2) = \frac{1}{8} (1 + e^{j\omega_1})^2 (1 + e^{j\omega_2})^2.$$

Thus from (3)

$$\tilde{\Psi}(\omega_1, \omega_2) = e^{-j\omega_1}(\frac{\sin \omega_1/2}{\omega_1/2})^2 e^{-j\omega_2}(\frac{\sin \omega_2/2}{\omega_2/2})^2.$$

The infinite sum $A(\omega_1, \omega_2)$ in this case becomes

$$A(\omega_1, \omega_2) = \sin^4 \frac{\omega_1}{2} \sin^4 \frac{\omega_2}{2} C(\frac{\omega_1}{2}, \frac{\omega_2}{2}).$$
where, using (26),
\[
C(x_1, x_2) = \sum_{m_1} \frac{1}{(x_1 + \pi m_1)^4} \sum_{m_2} \frac{1}{(x_2 + \pi m_2)^4} = \frac{11 + 2 \cos^2 x_1}{9 \sin^4 x_1} \frac{1 + 2 \cos^2 x_2}{\sin^4 x_2}.
\]

Thus, a biorthogonal wavelet base results, with
\[
A(\omega_1, \omega_2) = \frac{1}{9} (1 + 2 \cos^2 \frac{\omega_1}{2})(1 + 2 \cos^2 \frac{\omega_2}{2}) = \frac{(2 + \cos \omega_1)(2 + \cos \omega_2)}{9}.
\]

This gives the optimum analysis filter
\[
H(\omega_1, \omega_2) = \frac{1}{8} (1 + e^{-j\omega_1})^2(1 + e^{-j\omega_2})^2 \frac{2 + \cos \omega_1}{2 + \cos 2\omega_1} \frac{2 + \cos \omega_2}{2 + \cos 2\omega_2}.
\]

A rectangular grid is also constructed with this decomposition. However the scaling function is of higher order, i.e.
\[
\tilde{\psi}(u, v) = uv \chi(A) + (2 - u)v \chi(B) + u(2 - v) \chi(C) + (2 - u)(2 - v) \chi(D)
\]

where \(\chi(G)\) denotes the characteristic function of set \(G\):
\[
\chi(G) = \begin{cases} 
1 & (u, v) \in G \\
0 & \text{otherwise} 
\end{cases}
\]

and \(A, B, C, D\) are the domain rectangles in figure 2.

Figure 2: (a) Support of \(\tilde{\psi}(u, v)\) for algorithm 4 and (b) plot of \(\tilde{\psi}(u, v)\).
Algorithm 5

With $M = \text{diag}\{2, 2\}$ choose $e_3 = e_1 + e_2$ with $e_1, e_2$ given by (28). This corresponds to the most common in practice mesh obtained by regular triangulation of the parametric space, illustrated in figure 3.

The scaling function $\tilde{\psi}(u, v)$ is given by

$$\tilde{\psi}(u, v) = u\mathcal{X}(A) + v\mathcal{X}(B) + (u + 1 - v)\mathcal{X}(C) + (v + 1 - u)\mathcal{X}(D) + (2 - v)\mathcal{X}(E) + (2 - u)\mathcal{X}(F)$$

(28)

where $\mathcal{X}(G)$ denotes the characteristic function of set $G$:

$$\mathcal{X}(G) = \begin{cases} 1 & (u, v) \in G \\ 0 & \text{otherwise} \end{cases}$$

and $A, \ldots, F$ are domain triangles shown in figure 4 (Courant element [11]). This function is a base for all wire-frames defining a regular triangulation of $U$, as described above (a proof is given in Lemma 2 in the Appendix):

$$\begin{bmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{bmatrix} = \sum_{k,l} \begin{bmatrix} x_{kl} \\ y_{kl} \\ z_{kl} \end{bmatrix} \tilde{\psi}(u - k, v - l)$$

(29)

where $[x_{kl}, y_{kl}, z_{kl}]$ are the nodes of the wire-frame.

In this case

$$G(\omega_1, \omega_2) = \frac{1}{2}(1 + e^{j\omega_1})(1 + e^{j\omega_2})(1 + e^{j(\omega_1 + \omega_2)})$$

(30)
Figure 4: (a): The support of $\tilde{\psi}(u, v)$, for algorithm 5 (b) A plot of $\tilde{\psi}(u, v)$

or equivalently,

$$G(\omega_1, \omega_2) = 2e^{-j\omega_1/2} \cos \left(\frac{\omega_1}{2}\right)e^{-j\omega_2/2} \cos \left(\frac{\omega_2}{2}\right)e^{-j(\omega_1+\omega_2)/2} \cos \left(\frac{\omega_1 + \omega_2}{2}\right).$$

Thus, from (3)

$$\tilde{\Psi}(\omega_1, \omega_2) = e^{j\omega_1/2} \sin \frac{\omega_1}{2} e^{j\omega_2/2} \sin \frac{\omega_2}{2} e^{-j(\omega_1+\omega_2)/2} \sin \left(\frac{\omega_1 + \omega_2}{2}\right).$$

Consider now the infinite sum

$$A(\omega_1, \omega_2) \equiv \sum_{m_1, m_2} \tilde{\Psi}(\omega_1 + 2\pi m_1, \omega_2 + 2\pi m_2)\tilde{\Psi}(-\omega_1 - 2\pi m_1, -\omega_2 - 2\pi m_2)$$

$$= \sum_{m_1, m_2} \frac{\sin^2\left(\frac{\omega_1}{2} + \pi m_1\right) \sin^2\left(\frac{\omega_2}{2} + \pi m_2\right) \sin^2\left(\frac{\omega_1+\omega_2}{2} + \pi m_1 + \pi m_2\right)}{\left(\frac{\omega_1}{2} + \pi m_1\right)^2 \left(\frac{\omega_2}{2} + \pi m_2\right)^2 \left(\frac{\omega_1+\omega_2}{2} + \pi m_1 + \pi m_2\right)^2}$$

$$= \sin^2\left(\frac{\omega_1}{2}\right) \sin^2\left(\frac{\omega_2}{2}\right) \sin^2\left(\frac{\omega_1 + \omega_2}{2}\right) D\left(\frac{\omega_1}{2}, \frac{\omega_2}{2}\right)$$

(31)

where

$$D(x_1, x_2) = \sum_{m_1, m_2} \frac{1}{(x_1 + \pi m_1)^2 (x_2 + \pi m_2)^2 (x_1 + x_2 + \pi m_1 + \pi m_2)^2}.$$

(32)

In this case as well, it is clear that $A(\omega_1, \omega_2) \neq 1$, hence that the wavelet base defined by (30) cannot be orthonormal. From Lemma 1 in the Appendix and (31) it is specifically seen that

$$A(\omega_1, \omega_2) = \frac{1}{3} + \frac{2}{3} \cos\left(\frac{\omega_1 + \omega_2}{2}\right) \cos\left(\frac{\omega_1}{2}\right) \cos\left(\frac{\omega_2}{2}\right).$$
Since
\[
\cos \frac{\omega}{2} = e^{j\omega/2} \left( \frac{1 + e^{-j\omega}}{2} \right) = e^{j\omega/2} \left( \frac{1 + z^{-1}}{2} \right) = e^{-j\omega/2} \left( \frac{1 + z}{2} \right).
\]
\[
\sin \frac{\omega}{2} = e^{j\omega/2} \left( \frac{1 - e^{-j\omega}}{2j} \right) = e^{j\omega/2} \left( \frac{1 - z^{-1}}{2j} \right) = e^{-j\omega/2} \left( \frac{z - 1}{2j} \right).
\]

The above give as a function of \( z_k = e^{j\omega_k} \):
\[
A(\omega_1, \omega_2) = \frac{1}{3} + \frac{2}{3} \frac{1 + z_1^{-1}}{2} \frac{1 + z_2^{-1}}{2} \frac{1 + z_1 z_2}{2}
\]
(33)
and hence from (18), the optimal analysis filter is expressed by
\[
H(\omega_1, \omega_2) = G(-\omega_1, -\omega_2) \frac{A(\omega_1, \omega_2)}{A(2\omega_1, 2\omega_2)}
\]
where \( G(\omega_1, \omega_2) \) and \( A(\omega_1, \omega_2) \) are respectively given by (30) and (33).

**Algorithm 6**

A more complicated triangulation algorithm will be obtained if \( \xi_1, \xi_2 \) are given by (27) and \( \xi_3 = \xi_1 + \xi_2, \xi_4 = \xi_2 - \xi_1 \). The associated grid and basis function \( \hat{\psi}(u, v) \) are shown in figure 5. The implementation of the corresponding synthesis filter \( G(\omega_1, \omega_2) \) may be achieved using the sampling matrix \( M = \text{diag}\{2, 2\} \).

In this case, an algorithm similar to algorithm 5 is established, with each level of the pyramid constructed containing one fourth the number of nodes of the preceding level.

However, more interesting results are obtained if the quincunx matrix is used to define the wavelet basis:
\[
M = \begin{bmatrix}
-1 & 1 \\
1 & 1
\end{bmatrix}.
\]

In this case, each level of the resulting pyramid contains one half of the nodes of the preceding level. Since with this matrix \( M_\xi = \xi_4 \), \( M_\xi_4 = 2\xi_1 \), \( M_\xi_2 = \xi_3 \), \( M_\xi_3 = 2\xi_2 \), directly from (7):
\[
G(\omega_1, \omega_2) = \frac{\sqrt{2}}{4} \left( 1 + e^{j\omega_1} \right) \left( 1 + e^{j\omega_2} \right)
\]

18
and since $2\pi\mathbf{M}^{-T}q_1 = [\pi \pi]^T$, the necessary condition (15) is clearly satisfied. From (3):

$$\tilde{\Psi}(\omega_1, \omega_2) = e^{j\omega_1/2} \sin(\omega_1/2) e^{j\omega_2/2} \sin(\omega_2/2) e^{j(\omega_1+\omega_2)/2} \sin((\omega_1+\omega_2)/2) e^{j(\omega_1-\omega_2)/2} \sin((\omega_1-\omega_2)/2).$$

As before, the infinite sum (18) becomes,

$$A(\omega_1, \omega_2) = \sin^2 \frac{\omega_1}{2} \sin^2 \frac{\omega_2}{2} \sin^2 \frac{\omega_1 + \omega_2}{2} \sin^2 \frac{\omega_1 - \omega_2}{2} E\left(\frac{\omega_1}{2}, \frac{\omega_2}{2}\right).$$

where

$$E(x_1, x_2) = \sum_{m_1} \sum_{m_2} \frac{1}{(x_1 + \pi m_1)^2(x_2 + \pi m_2)^2(x_1 + x_2 + \pi m_1 + \pi m_2)^2(x_1 - x_2 + \pi m_1 - \pi m_2)^2}.$$
IV EXPERIMENTAL RESULTS

The optimal decomposition filters $H, G$ obtained in the previous section are used to construct a multiresolution representation of the original surface. This is illustrated in figure 7.

In the upper branch a series of successively lower resolution surfaces $x_1, x_2, \ldots, x_n$ are constructed by filtering with $H$ and down-sampling. Specifically if $z_k = e^{j\omega_k}, \ k = 1, 2$

$$H(-\omega_1, -\omega_2) = \frac{\sum_{m\geq 0, n} d_{mn} x_{k-m,l-n} z_{k-1}^{-m} z_{l}^{-n}}{1 - \sum_{k\geq 0, l} d_{kl} x_{k-1}^{-m} z_{l}^{-n}}$$

then the filtered wire-frame $[\tilde{x}_{kl}, \tilde{y}_{kl}, \tilde{z}_{kl}]$ is obtained recursively by:

$$
\begin{bmatrix}
\tilde{x}_{kl} \\
\tilde{y}_{kl} \\
\tilde{z}_{kl}
\end{bmatrix} =
\begin{bmatrix}
\sum_{m\geq 0, n} d_{mn} \tilde{x}_{k-m,l-n} \\
\sum_{m\geq 0, n} d_{mn} \tilde{y}_{k-m,l-n} \\
\sum_{m\geq 0, n} d_{mn} \tilde{z}_{k-m,l-n}
\end{bmatrix} +
\begin{bmatrix}
\sum_{m,n} p_{mn} x_{k-m,l-n} \\
\sum_{m,n} p_{mn} y_{k-m,l-n} \\
\sum_{m,n} p_{mn} z_{k-m,l-n}
\end{bmatrix}
$$

while down-sampling consists of the multiplication of index $[k \ l]^T$ by the sampling matrix $M$. Also, error
surfaces $e_1, \ldots, e_n$ are constructed at this step. These correspond to the error for each wire-frame node due to scale reduction at each level. The lowest resolution level surface $x_n$ and error surfaces may be used to reconstruct the original signal as the lower branch in fig. 7 illustrates. Since the analysis filters selected minimize the error $E_s$ in (20) for arbitrary resolution $s$, the combination of $x_n$ with as many of the error images $e_n, e_{n-1}, \ldots, e_{n_q}$ as may be born by a slower communication channel will produce an optimal, in the sense discussed, lower resolution version of the original image for transmission via this channel. Optimal progressive coding will likewise be achieved if $x_n$ is gradually combined with error images until the desired quality is attained.

In order to implement the analysis filters $H$ in algorithms 4 and 5, the methodology in [32] was used. This requires the analysis of $A(\omega_1, \omega_2)$ to causal and anti-causal factors.

$$A(\omega_1, \omega_2) = K(z_1, z_2)K(z_1^{-1}, z_2^{-1})$$

where $z_k = e^{j\omega_k}, \ k = 1, 2$. The coefficients of $K$ were found using the non-symmetric half plane factor-
ization algorithm in [33]. For algorithm 4 this resulted in
\[
K(z_1, z_2) = (1 + 0.26795z_1)(1 + 0.26795z_2).
\]
for algorithm 5, the resulting coefficients of \(K(z_1, z_2)\) are given in table 1, and for algorithm 6 in table 2.

<table>
<thead>
<tr>
<th>factor</th>
<th>coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>(z_2)</td>
<td>2.350061</td>
</tr>
<tr>
<td>(z_1z_2^{-1})</td>
<td>-0.055694</td>
</tr>
<tr>
<td>(z_1)</td>
<td>0.356663</td>
</tr>
<tr>
<td>(z_1z_2)</td>
<td>0.422662</td>
</tr>
</tbody>
</table>

Table 1: The filter \(K(z_1, z_2)\) for algorithm 5

<table>
<thead>
<tr>
<th>factor</th>
<th>coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>(z_2)</td>
<td>12.063366</td>
</tr>
<tr>
<td>(z_2^{-3})</td>
<td>0.939145</td>
</tr>
<tr>
<td>(z_1z_2^{-3})</td>
<td>2.741844</td>
</tr>
<tr>
<td>(z_1z_2^{-1})</td>
<td>0.860102</td>
</tr>
<tr>
<td>(z_1)</td>
<td>7.355003</td>
</tr>
<tr>
<td>(z_2)</td>
<td>0.002453</td>
</tr>
<tr>
<td>(z_1z_2)</td>
<td>0.082888</td>
</tr>
</tbody>
</table>

Table 2: Algorithm 6: (a) Factorization \(A(\omega_1, \omega_2) = K(z_1, z_2)K(z_1^{-1}, z_2^{-1})\) (b) Factorization \(A(\omega_1 - \omega_2, \omega_1 + \omega_2) = L(z_1, z_2)L(z_1^{-1}, z_2^{-1})\)

A simple encoding scheme was used to compress \(x_n\) and \(e_1, \ldots, e_n\), consisting of DPCM prediction, quantization and entropy coding using a fixed Huffman table. Quantization tables adapted to a Laplacian distribution were applied to quantization of \(e_1, \ldots, e_n\), while uniform quantization is used for \(x_n\).

A set of experiments was conducted to evaluate the results obtained in the previous section. The experiments were performed using synthetic and natural data sets of various modalities. Two of the data-sets “head” and “brain” are surfaces obtained from segmentation of 3D MRI data. The wire-frames corresponding to these surfaces were constructed by uniform sampling of range data extracted from the MRI volume. The wire-frames consist of \(256 \times 256\) nodes which are originally stored as floating point number triplets. These surfaces were decomposed to multiple resolution levels as shown in figures 8, 9. The test data “fiber”
were obtained by regular triangulation (256 × 256 nodes) of 3D confocal laser scanning microscope images, and represent the surface geometry from a very small region (200 × 200 × 10 microns) of paper fibers [36]. Surface rendering of this surface at various resolutions is shown in figure 11.

Finally the data set “ocean” is obtained from satellite data on the earth surface. This data set consists of a color image \( s(\phi, \theta) \), where \(-\pi < \phi \leq \pi, -\pi/2 < \theta \leq \pi/2\) are azimuth angles, and each pixel is a pseudocolor representation of the ocean phytoplankton concentration \(^1\). By a regular triangulation (2048 × 1024 nodes) of this data, we obtain a wire-frame in the RGB (red, green, blue) space i.e.

\[
\begin{bmatrix}
  r(\phi, \theta) \\
  g(\phi, \theta) \\
  b(\phi, \theta)
\end{bmatrix}
= \sum_{k,l}
\begin{bmatrix}
  r_{kl} \\
  g_{kl} \\
  b_{kl}
\end{bmatrix}
\tilde{\psi}(\phi - k, \theta - l)
\]

The color surfaces obtained in various resolution levels are then mapped on a sphere (see figure 12).

All six of the optimal algorithms were tested on the above data so as to compare their efficiency for the progressive coding of 3D meshes. Further, in order to evaluate their performance for pyramidal compression against nonoptimal algorithms, they were also implemented using the nonoptimal analysis filter \( H(\omega_1, \omega_2) = 1 \), which corresponds to simple subsampling of the node lattices from one level to the next. This approach is used by interpolating subdivision techniques [46]. Also, the reconstruction error obtained by applying the algorithm in [23] (using a polyhedral subdivision scheme) is reported for the case of 4:1 and 16:1 scale reduction. It is seen that the nonoptimal algorithm produces significantly higher errors. In fact, the nonoptimal algorithm produces at scale reductions 4:1, 8:1 errors comparable to those of the optimal algorithms at scale reductions of 8:1 and 16:1 respectively.

In table 3 the reconstruction errors for the optimal filters corresponding to algorithms 2 to 5, are given for the various data sets.

The reconstruction errors for the optimal filters corresponding to algorithms 1 and 6 are given in table 4.

In table 5 we compare the efficiency (in terms of the resulting bits required for coding of each vertex) of various algorithms in lossless image compression. The pyramids for algorithms 1-3 were transformed into reduced pyramids (i.e. pyramids with number of data equal to the number of data of the original mesh) using the method described in [44]. As it is clear from this table, quincunx or hexagonal sampling lead to

\(^1\)This data is from NASA SeaWiFS project (http://seawifs.gsfc.nasa.gov/SEAWIFS/CZCS_DATA/global_full/ november78_june86.chlor_global.gif)
A comparison of the above 6 algorithms on the basis of their performance as reflected in tables 3-5, shows that

1. Algorithm 1, based on the quincunx matrix box splines, offers the best compression in terms of bits/vertex required for lossless transmission.

2. In terms of higher similarity of the original to the reduced-scale mesh versions, best results are achieved by algorithms 1 and 6 for a reduction of 2:1 and algorithms 4 and 6 for a 4:1 scale reduction. Of the three algorithms, 1 and 6 are based on quincunx box splines defined in this paper, while 4 is based on the traditional (separable) box splines in [11, 26].
3. Although some algorithms are clearly simpler than the others, no significant differences in computing time were noticed in the above experiments.

V CONCLUSIONS

Optimal mechanisms were developed and tested for the hierarchical decomposition of wire-frame surfaces generated by box-splines. A family of box-splines with compact support was used, generated by arbitrary sampling matrices with integer eigenvalues. For each such box-spline the optimal node configuration was determined for each level of the hierarchical wire-frame decomposition. Criterion of optimality was the minimization of the variance of the error difference between the original surface and its version at the lower resolution level. Several such combinations of box-spline generated meshes and the corresponding optimal node lattice sequences were examined in detail. Results in the area of hierarchical 3D object coding were evaluated experimentally. A typical application is the formation of a series of replicas of the original image in lower resolutions for their transmission via slower channels or for the realization of progressive coding in which the original low resolution gradually increases until the desired quality is achieved. Many other applications of the proposed scheme may be also found in the the computer graphics and the computer vision area.

VI Acknowledgment

This work was supported by the European project PANORAMA (Package for New Autostereoscopic Multiview Systems and Applications, ACTS project 092).

VII APPENDIX

Lemma 1:

If $x_1, x_2, y$ are real then

\[ \sum_{m_1} \sum_{m_2} \frac{1}{(x_1 + \pi m_1)(x_2 + \pi m_2)(y + \pi m_1 + \pi m_2)} = \frac{1}{\sin(x_1 + x_2 - y) \sin x_1 \sin x_2 \sin y}. \]  

(A1)
(ii) 
\[ \sum_{m_1} \sum_{m_2} \frac{1}{(x_1 + \pi m_1)^2(x_2 + \pi m_2)^2(x_1 + x_2 + \pi m_1 + \pi m_2)^2} = \frac{1}{3} (\cot(x_1 + x_2) - \cot x_1 - \cot x_2) - \cot x_1 \cot x_2 \cot(x_1 + x_2) \sin x_1 \sin x_2 \sin(x_1 + x_2). \] (A2)

(iii) 
\[ S(x_1, x_2) = \sum_{m_1} \sum_{m_2} \frac{1}{(x_1 + \pi m_1)^3(x_2 + \pi m_2)^3(x_1 + x_2 + \pi m_1 + \pi m_2)^2} = \frac{R(x_1, x_2)}{\sin x_1 \sin x_2 \sin(x_1 + x_2)}. \] (A3)

where 
\[ R(x_1, x_2) = \frac{1}{3} \cot x_1 \cot x_2 \cot(x_1 + x_2) + \frac{7}{15} \cot(x_1 + x_2) - \frac{4}{15} \cot x_1 - \frac{4}{15} \cot x_2 - \frac{2}{3} \csc^2 x_1 - \frac{2}{3} \csc^2 x_2 + \frac{\cot(x_1 + x_2)}{\sin^2 x_1 \sin^2 x_2} + \frac{1}{3} \cot x_1 + \frac{1}{3} \cot x_2 + \frac{1}{3} \cot x_1 + \frac{1}{3} \cot x_2. \]

(iv) 
\[ E(x_1, x_2) = \sum_{m_1} \sum_{m_2} \frac{1}{(x_1 + \pi m_1)^2(x_2 + \pi m_2)^2(x_1 + x_2 + \pi m_1 + \pi m_2)^2} = \frac{1}{4} (S(x_1, x_2) + S(x_1, -x_2)). \] (A4)

where \(S(x_1, x_2)\) is defined by (A3).

Proof

From the well known identity \([39]\) 
\[ \sum_{m_1} \frac{1}{a + \pi m_1} = \cot a \]

we obtain 
\[ \cot a - \cot b = \sum_{m_1} \frac{1}{(a + \pi m_1)(b + \pi m_1)}. \]
With $a = x_1$, $b = y + \pi m_2$ we find

\[
\frac{\cot x_1 - \cot y}{y - x_1 + \pi m_2} = \sum_{m_1} \frac{1}{(x_1 + \pi m_1)(y + \pi m_1 + \pi m_2)}.
\]

Therefore, the left hand side of (A1) equals

\[
\sum_{m_2} \frac{\cot x_1 - \cot y}{(x_2 + \pi m_2)(y - x_1 + \pi m_2)} = \frac{\cot x_1 - \cot y}{y - x_1 - x_2} \sum_{m_2} \left( \frac{1}{x_2 + \pi m_2} - \frac{1}{y - x_1 + \pi m_2} \right) = \frac{\cot x_1 - \cot y}{y - x_1 - x_2} (\cot x_2 - \cot (y - x))
\]

whence using the identity

\[
\cot x \pm \cot y = \pm \frac{\sin(x \pm y)}{\sin x \sin y}
\]

we find that the left-hand side of (A1) equals

\[
-\frac{\sin(x_1 - y)}{\sin x_1 \sin(y - x_1 - x_2)} - \frac{\sin(x_2 + x_1 - y)}{\sin x_2 \sin(y - x_1)} = \frac{\sin(x_1 + x_2 - y)}{(x_1 + x_2 - y) \sin x_1 \sin x_2 \sin y}
\]

and part (i) of the Lemma follows.

To prove part (ii) it suffices to partially differentiate both sides of (A1) by $x_1, x_2, y$ and set $y = x_1 + x_2$.

The partial derivative of the left-hand side of (A2) is then equal to the left hand-side of (A2), and the derivative of the right-hand side of (A1) is the right-hand side of (A2).

Similarly (A3) is obtained by differentiating both sides of (A1) twice with respect to $x_1, x_2$, and once with respect to $y$, and setting $y = x_1 + x_2$.

To prove (iv) simply note that

\[
\sum_{m_1} \sum_{m_2} \frac{1}{(x_1 + \pi m_1)^3(x_2 + \pi m_2)^3} = \frac{1}{(x_1 + \pi m_1)^3(x_2 + \pi m_2)^3} - \frac{1}{(x_1 + \pi m_1 - \pi m_2)^2} = -S(x_1, -x_2). \tag{A5}
\]

Thus, subtracting by parts (A3) ad (A5) yields:

\[
S(x_1, x_2) + S(x_1, -x_2) = \sum_{m_1} \sum_{m_2} \frac{1}{(x_1 + \pi m_1)^3(x_2 + \pi m_2)^3} \left[ \frac{1}{(x_1 + x_2 + \pi m_1 + \pi m_2)^2} - \frac{1}{(x_1 - x_2 + \pi m_1 - \pi m_2)^2} \right] = \]
\[
= \sum_{m_1} \sum_{m_2} (x_1 + \pi m_1) (x_2 + \pi m_2) (-4)(x_1 + \pi m_1)(x_2 + \pi m_2)
\]

\[
(\mathcal{m} x_2 + \pi m_2)^3 (\mathcal{m} x_1 + \pi m_1)^3 (\mathcal{m} x_2 + \pi m_2)^2 (\mathcal{m} x_1 - x_2 + \pi m_1 - \pi m_2)^2
\]

which proves (A4) and the lemma.

Lemma 2:

The hat function (28) is a base for all wire-frames defined on a regular triangulation (figure 3) of the parametric space.

Proof:

Let \((u, v)\) be a point inside the domain triangle with vertex indices \((k, l), (k, l + 1)\) and \((k + 1, l + 1)\).

Then, using (28), (29) becomes:

\[
P(u, v) = \begin{bmatrix}
x(u, v) \\
y(u, v) \\
z(u, v)
\end{bmatrix} = (1 - u)P_{k-1,l-1} + vP_{k,l} + (u - v)P_{k,l-1}
\]

(A1)

Since \(1 - u + v + u - v = 1\), \(P(u, v)\) is a barycentric combination of the wire-frame nodes \(P_{k-1,l-1}, P_{k,l}\)

and \(P_{k,l-1}\), and therefore lies on the plane defined by these nodes [34].

References


Figure 8: “head” (a) original surface (b) 4:1 scale surface (c) 16:1 scale surface (d) reconstructed surface (compression 8:1), algorithm 5 ; (e) 2:1 scale surface (f) 8:1 scale surface, algorithm 6
Figure 9: “brain” (a) original surface (b) 4:1 scale surface (c) 16:1 scale surface (d) reconstructed surface (compression 8:1), algorithm 2

Figure 10: “brain” (a) 2:1 scale surface (b) 4:1 scale surface (c) 8:1 scale surface (d) reconstructed surface (compression 8:1), algorithm 6
Figure 11: “fiber” (a) original surface (b) 4:1 scale surface (algorithm 5) (c) 16:1 scale surface (two steps of algorithm 5) (d) reconstructed surface (compression 8:1) (e) 2:1 scale surface (algorithm 6) (f) 8:1 scale surface (3 steps of algorithm 6)
Figure 12: “ocean” (a) original surface (b) 4:1 scale surface (c) 16:1 scale surface (d) reconstructed surface (compression 8:1), algorithm 5